AMENDMENTS TO THE CLAIMS:

This listing of claims will replace all prior versions, and listings, of claims in the application:

LISTING OF CLAIMS:

Claims 1-105. (Canceled)

106. (New) A method for transmitting data over a network to at least one client having a latency time to initiate transmission of said data to the client, including the steps of:

generating M anti-latency data streams containing at least a leading portion of data for receipt by the client, wherein said data is fragmented into K segments each requiring a time T to transmit over the network, and wherein each of the M anti-latency data streams contains substantially identical data having J segments that are repeated continuously within said anti-latency data stream, and each successive anti-latency data stream is staggered by an anti-latency time interval $\geq T$; and

generating N interactive data streams containing at least a remaining portion of said data for the client to merge into after receiving at least a portion of an anti-latency data stream, wherein each of the N interactive data streams repeats continuously within said interactive data stream, and each successive interactive data stream is staggered by an interactive time interval;

wherein: J, K, M and N are integers,

T is a length of time, and

$M = N = J = \sqrt{R/T}$,

where R is the length of time required to transmit said data over the network.

107. (New) A method for transmitting data over a network to at least one client having a latency time to initiate transmission of said data to the client, including the steps of:

generating M anti-latency data streams containing at least a leading portion of data for receipt by the client, wherein said data is fragmented into K segments each requiring a time T to transmit over the network, wherein each anti-latency data stream includes:

a leading data stream containing at least one leading segment of the leading portion of said data being repeated continuously within the leading data stream; and

a plurality of finishing data streams each having J segments, each of the finishing data streams containing the rest of the leading portion of said data; and being repeated continuously within said finishing data stream, and wherein each successive finishing data stream is staggered by an anti-latency time interval $\geq T$; and

generating N interactive data streams containing at least a remaining portion of said data for the client to merge into after receiving at least a portion of an anti-latency data stream, wherein each of the N interactive data streams is repeated continuously within said interactive data stream, and each successive interactive data stream is staggered by an interactive time interval;

wherein: J, K, M and N are integers,

T is a length of time,

$M = (J/2) + 1$, and

$J = \sqrt{2}K$.

108. (New) A method for transmitting data over a network to at least one client having a latency time to initiate transmission of said data to the client, including the steps of:

generating M anti-latency data streams containing at least a leading portion of data for receipt by the client, wherein said data is fragmented into K segments each requiring a time T to transmit over the network, and wherein the anti-latency data streams are generated such that an m^{th} anti-latency data stream has F_m segments, wherein F_m is an m^{th} Fibonacci number; and the F_m segments are repeated continuously within the m^{th} anti-latency data stream; and

generating N interactive data streams containing at least a remaining portion of said data for the client to merge into after receiving at least a portion of an anti-latency data stream, wherein each of the N interactive data streams repeats continuously within said interactive data stream, and each successive interactive data stream is staggered by an interactive time interval = KT/N;

wherein m starts from 4 and the repeating 1^{st}, 2^{nd}, and 3^{rd} anti-latency data streams have the following configuration:

1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1
2	3	2	3	2	3	2	3	2	3	2	3	2	3	2	3	2	3	2	3	2	3	2	3	2	3	2	3	2	3	2	3	2	3
4	5	6	7	4	5	6	7	4	5	6	7	4	5	6	7	4	5	6	7	4	5	6	7	4	5	6	7	4	5	6	7	4	5

and wherein: K, M and N are integers, and

T is a length of time.

109. (New) A method for transmitting data over a network to at least one client having a latency time to initiate transmission of said data to the client, including the steps of:

generating *M* anti-latency data streams containing at least a leading portion of data for receipt by the client, wherein said data is fragmented into *K* segments each requiring a time *T* to transmit over the network, and wherein in the *M* anti-latency data streams the leading portion of said data contains 1 to *J* leading labeled data segments, and the leading data segments are distributed in the *M* anti-latency data streams such that a j^{th} leading segment is repeated by an anti-latency time interval $\leq jT$ within the anti-latency data streams; and

generating *N* interactive data streams containing at least a remaining portion of said data for the client to merge into after receiving at least a portion of an anti-latency data stream, wherein each of the *N* interactive data streams repeats continuously within said interactive data stream, and each successive interactive data stream is staggered by an interactive time interval = *KT/N*;

wherein: *J, K, M and N* are integers,

T is a length of time,

$J = K/N$, and $M \geq \sum_{j=1}^{j=J}(\frac{1}{j})$ and $J = \frac{K}{N}$.

110. (New) A method for transmitting data over a network to at least one client having a latency time to initiate transmission of said data to the client, including the steps of:

generating M anti-latency data streams containing at least a leading portion of data for receipt by the client, wherein said data is fragmented into K segments each requiring a time T to transmit over the network, and wherein in the M anti-latency data streams the leading portion of said data contains 1 to J leading labeled data segments, and the leading data segments are distributed in the M anti-latency data streams such that a j^{th} leading segment is repeated by an anti-latency time interval $\leq jT$ within the anti-latency data streams; and

generating N interactive data streams containing at least a remaining portion of said data for the client to merge into after receiving at least a portion of an anti-latency data stream, wherein each of the N interactive data streams repeats continuously within said interactive data stream, and each successive interactive data stream is staggered by an interactive time interval = KT/N;

wherein: J, K, M and N are integers,

T is a length of time,

$J = K/N$,

and wherein six of the M anti-latency data streams containing the leading data segments are arranged as follows:

1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1
2	4	2	8	2	4	2	16	2	4	2	8	2	4	2	32	2	4	2	8	2	4	2	16	2	4	2	8	2	4	2	64	2	4
3	6	9	3	12	42	3	6	18	3	24	9	3	6		3	12	48	3	6	9	3	36	39	3	6	18	3	12	9	3	6		3
5	10	15	25		5	20				5	10	30		35	5	40	15			5	10	45	50		5	20	55	25	60	5	10	15	
7	14	21	11	13	17	19	7	28	22	23	26	27	11	7	14	29	13	31	33	34	7	17	21	11	19	37	31	7	14	13	22	33	23
38	41		43	44		46	47	49				51	52				53		54			56	57		58		59		61		62	63	

wherein blank segments contain any data.

111. (New) A method for transmitting data over a network to at least one client having a latency time to initiate transmission of said data to the client, including the steps of:

generating *M* anti-latency data streams containing at least a leading portion of data for receipt by the client, wherein said data is fragmented into *K* segments each requiring a time *T* to transmit over the network, and wherein the M anti-latency data streams contain the leading portion of said data, and further include two batches of data streams comprising a first set of anti-latency data streams and a second set of anti-latency data streams, such that

the first anti-latency data streams have *A* first anti-latency data streams from I to A, wherein an a^{th} anti-latency data stream has F_a segments, where F_a is an a^{th} Fibonacci number, and the F_a segments are repeated continuously within the a^{th} first anti-latency data stream

the second anti-latency data streams have *B* second anti-latency data streams wherein each of the B second anti-latency data streams contains substantially identical data repeated continuously within said second anti-latency data stream, and wherein each successive second anti-latency data stream is staggered by a coarse-jump frame period, such that the client can perform a coarse-jump function when the client is connected to a second anti-latency data steam; and

generating *N* interactive data streams containing at least a remaining portion of said data for the client to merge into after receiving at least a portion of an anti-latency data stream;

wherein: *A, B, K, M and N* are integers, and

T is a length of time.

112. (New) The method of claim 111, further including the steps of:

connecting the client to at least the a^{th} and $(a+1)^{th}$ first anti-latency data streams when the client raises a request for said data;

buffering the data in at least the a^{th} and $(a+1)^{th}$ first anti-latency data streams in the client;

subsequently connecting the client to successive first anti-latency data streams; and

repeating the previous steps until all data in the A first anti-latency data streams is received by the client.

113. (New) The method of claim 112, further including the steps of:

connecting the client to any one of the B second anti-latency data streams after all data in the first anti-latency data streams is received by the client; and

connecting the client to anyone of the N interactive data streams after all data in the connected B second anti-latency data stream is received by the client.

114. (New) The method of claim 111, wherein each of the N interactive data streams contains the whole set of said data having K segments.

115. (New) The method of claim 111, wherein each of the N interactive data streams contains the remaining portion of said data only.

116. (New) The method of claim 111, wherein said coarse-jump frame period includes E data segments, and $F_a \geq 2E$.

117. (New) The method of claim 111, wherein a starts from 1.

118. (New) The method of claim 111, wherein a starts from 4 and the repeating first, second, and third anti-latency data streams have the following configuration:

1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1
2	3	2	3	2	3	2	3	2	3	2	3	2	3	2	3	2	3	2	3	2	3	2	3	2	3	2	3	2	3	2	3	2	3
4	5	6	7	4	5	6	7	4	5	6	7	4	5	6	7	4	5	6	7	4	5	6	7	4	5	6	7	4	5	6	7	4	5

119. (New) A method for transmitting data over a network to at least one client having a latency time to initiate transmission of said data to the client, including the steps of:

generating M anti-latency data streams containing at least a leading portion of data for receipt by the client, wherein said data is fragmented into K segments each requiring a time T to transmit over the network, and wherein the M anti-latency data streams contain the leading portion of said data, and further include two batches of data streams comprising a first set of anti-latency data streams and a second set of anti-latency data streams, such that

the first anti-latency data streams have A first anti-latency data streams from 1 to A, wherein an a^{th} anti-latency data stream has F_a segments, and F_a is an a^{th}

Fibonacci number; and the F_a segments are repeated continuously within the a^{th} first anti-latency data stream, and

the second anti-latency data streams have *B* second anti-latency data streams including a leading data stream containing at least one leading segment of the leading portion of said data being repeated continuously within the leading data stream; and a plurality of finishing data streams, each of the finishing data streams containing the rest of the leading portion of said data; and being repeated continuously within said finishing data stream, wherein each successive finishing data steam is staggered by a coarse-jump frame period such that the client can perform a coarse-jump interactive function when the client is connected to a second anti-latency data stream; and

generating *N* interactive data streams containing at least a remaining portion of said data for the client to merge into after receiving at least a portion of an anti-latency data stream;

wherein: *A, B, K, M and N* are integers, and

T is a length of time.

120. (New) The method of claim 119, further including the steps of:

connecting the client to at least the a^{th} and $(a+1)^{th}$ first anti-latency data streams when the client raises a request for said data;

buffering the data in at least the a^{th} and $(a+1)^{th}$ first anti-latency data streams in the client;

subsequently connecting the client to successive first anti-latency data streams; and

repeating the previous steps until all data in the *A* first anti-latency data streams is received by the client.

121. (New) The method of claim 120, further including the steps of:

connecting the client to the leading data stream after all data in the first anti-latency data streams is received by the client;

subsequently connecting the client to any one of the finishing data streams; and

connecting the client to any one of the *N* interactive data streams after all data in the *B* second anti-latency data streams is received by the client.

122. (New) The method of claim 119, wherein each of the *N* interactive data steams contains the whole set of said data having *K* segments.

123. (New) The method of claim 119, wherein each of the *N* interactive data streams contains the remaining portion of said data only.

124. (New) The method of claim 119, wherein said coarse-jump frame period includes *E* data segments, and $F_a \geq 2E$.

125. (New) The method of claim 119, wherein *a* starts from 1.

126. (New) The method of claim 119, wherein *a* starts from 4 and the repeating first, second, and third data streams of the *A* first anti-latency data streams have the following configuration:

1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1
2	3	2	3	2	3	2	3	2	3	2	3	2	3	2	3	2	3	2	3	2	3	2	3	2	3	2	3	2	3	2	3	2	3
4	5	6	7	4	5	6	7	4	5	6	7	4	5	6	7	4	5	6	7	4	5	6	7	4	5	6	7	4	5	6	7	4	5

127. (New) A method for transmitting data over a network to at least one client having a latency time to initiate transmission of said data to the client, including the steps of:

generating *M* anti-latency data streams containing at least a leading portion of data for receipt by the client, wherein said data is fragmented into *K* segments each requiring a time *T* to transmit over the network, and wherein the M anti-latency data streams contain the leading portion of said data, and further include two batches of data streams comprising a first set of anti-latency data streams and a second set of anti-latency data streams, such that

the first anti-latency data streams have *A* first anti-latency data streams, wherein, I. the A first anti-latency data streams contains 1 to *C* first data segments; and II. the first data segments are distributed in the *A* first anti-latency data streams such that an c^{th} leading segment is repeated by an anti-latency time interval $\leq cT$ within the *A* first anti-latency data streams, and

the second anti-latency data streams have *B* second anti-latency data streams, wherein each of the *B* second anti-latency data streams contains substantially identical data repeated continuously within said second anti-latency

data stream, and wherein each successive second anti-latency data stream is staggered by a coarse-jump frame period; such that the client can perform a coarse-jump interactive function when the client is connected to a second anti-latency data stream; and

generating *N* interactive data streams containing at least a remaining portion of said data for the client to merge into after receiving at least a portion of an anti-latency data stream;

wherein: *A, B, C, K, M and N* are integers, and

T is a length of time.

128. (New) The method of claim 127, further including the steps of:

connecting the client to all of the *A* first anti-latency data streams when the client raises a request for said data; and

buffering data in the *A* first anti-latency data streams in the client until all data in the *A* first anti-latency data streams is received by the client.

129. (New) The method of claim 128, further including the steps of:

connecting the client to any one of the *B* second anti-latency data streams after all data in the first anti-latency data streams is received by the client; and connecting the client to anyone of the *N* interactive data streams after all data in the connected *B* second anti-latency data stream is received by the client.

130. (New) The method of claim 127, wherein each of the *N* interactive data streams contains the whole set of said data having *K* segments.

131. (New) The method of claim 127, wherein each of the *N* interactive data streams contains the remaining portion of said data only.

132. (New) The method of claim 127, wherein said coarse-jump frame period includes *E* data segments, and $A \geq \sum_{c=1}^{c=E} (\frac{1}{c})$.

133. (New) The method of claim 127, wherein six of the *A* first anti-latency data streams are arranged as follows:

1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1
2	4	2	8	2	4	2	16	2	4	2	8	2	4	2	32	2	4	2	8	2	4	2	16	2	4	2	8	2	4	2	66	2	4
3	6	9	3	12	42	3	6	18	3	24	9	3	6		3	12	48	3	6	9	3	36	39	3	6	18	3	12	9	3	6		3
5	10	15	25		5	20				5	10	30		35	5	40	15			5	10	45	50		5	20	55	25	60	5	10	15	
7	14	21	11	13	17	19	7	28	22	23	26	27	11	7	14	29	13	31	33	34	7	17	21	11	19	37	31	7	14	13	22	33	23
38	41		43	44		46	47	49				51	52				53		54			56	57		58		59		61		62	63	

wherein blank segments contain any data.

134. (New) A method for transmitting data over a network to at least one client having a latency time to initiate transmission of said data to the client, including the steps of:

generating *M* anti-latency data streams containing at least a leading portion of data for receipt by the client, wherein said data is fragmented into *K* segments each requiring a time *T* to transmit over the network, and wherein the M anti-latency data streams contain the leading portion of said data, and further include two batches of

data streams comprising a first set of anti-latency data streams and a second set of anti-latency data streams, such that

the first anti-latency data streams have *A* first anti-latency data steams, wherein the *A* first anti-latency data streams contain 1 to *C* first data segments, and the first data segments are distributed in the *A* first anti-latency data streams such that an c^{th} leading segment is repeated by an anti-latency time interval $\leq cT$ within the *A* first anti-latency data streams, and

the second anti-latency data streams have *B* second anti-latency data stream including a leading data stream containing at least one leading segment of the leading portion of said data being repeated continuously within the leading data stream, and a plurality of finishing data streams, each of the finishing data streams connecting the rest of the leading portion of said data, and being repeated continuously with said finishing data stream, and wherein each successive finishing data stream is staggered by a coarse-jump frame period such that the client can perform a coarse-jump interactive function when the client is connected to a second anti-latency data stream; and

generating *N* interactive data streams containing at least a remaining portion of said data for the client to merge into after receiving at least a portion of an anti-latency data stream;

wherein: *A, B, C, K, M and N* are integers, and

T is a length of time.

135. (New) The method of claim 134, further including the steps of:

connecting the client to all of the *A* first anti-latency data streams when the client raises a request for said data; and

buffering data in the *A* first is anti-latency data streams in the client until all data in the *A* first anti-latency data streams is received by the client.

136. (New) The method of claim 135, further including the steps of:

connecting the client to the leading data stream of the *B* second anti-latency data streams after all data in the first anti-latency data streams is received by the client;

subsequently connecting the client to any one of the finishing data streams; and

connecting the client to any one of the *N* interactive data streams after all data in the second anti-latency data stream is received by the client.

137. (New) The method of claim 134, wherein each of the *N* interactive data streams contains the whole set of said data having *K* segments.

138. (New) The method of claim 134, wherein each of the *N* interactive data streams contains the remaining portion of said data only.

139. (New) The method of claim 134, wherein said coarse-jump frame period includes E data segments, and $A \geq \sum_{c=1}^{c=E} (\frac{1}{c})$.

140. (New) The method of claim 136, wherein six of the A first anti-latency data steams are arranged as follows:

1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1
2	4	2	8	2	4	2	16	2	4	2	8	2	4	2	32	2	4	2	8	2	4	2	16	2	4	2	8	2	4	2	64	2	4
3	6	9	3	12	42	3	6	18	3	24	9	3	6		3	12	48	3	6	9	3	36	39	3	6	18	3	12	9	3	6		3
5	10	15	25		5	20				5	10	30		35	5	40	15			5	10	45	50		5	20	55	25	60	5	10	15	
7	14	21	11	13	17	19	7	28	22	23	26	27	11	7	14	20	13	31	33	34	7	17	21	11	19	37	31	7	14	13	22	33	23
38	41		43	44		46	47	49				51	52				53		54			56	57		58		59		61		62	63	

wherein blank segments contain any data.

141. (New) A method for transmitting data over a network to at least one client including the step of generating M anti-latency data streams from 1 to M, wherein an m^{th} anti-latency data stream has F_m segments, and F_m is an m^{th} Fibonacci number; and said F_m segments are repeated continuously within the m^{th} anti-latency data stream, and wherein m starts from 4 and the repeating first, second, and third anti-latency data streams have the following configuration:

1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1
2	3	2	3	2	3	2	3	2	3	2	3	2	3	2	3	2	3	2	3	2	3	2	3	2	3	2	3	2	3	2	3	2	3
4	5	6	7	4	5	6	7	4	5	6	7	4	5	6	7	4	5	6	7	4	5	6	7	4	5	6	7	4	5	6	7	4	5

142. (New) A method for transmitting data over a network to at least one client, said data being fragmented into K segments each requiring a time T to transmit over the network, including the steps of: generating M anti-latency data streams containing 1 to K anti-latency data segments, wherein the anti-latency data segments are distributed in the M anti-latency data streams such that an k^{th} leading

segment is repeated by an anti-latency time interval $\leq kT$ within the anti-latency data streams, and wherein six of the M anti-latency data streams containing the leading data segments are arranged as follows: :

1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1
2	4	2	8	2	4	2	16	2	4	2	8	2	4	2	32	2	4	2	8	2	4	2	16	2	4	2	8	2	4	2	64	2	4
3	6	9	3	12	42	3	6	18	3	24	9	3	6		3	12	48	3	6	9	3	36	39	3	6	18	3	12	9	3	6		3
5	10	15	25		5	20				5	10	30		35	5	40	15			5	10	45	50		5	20	55	25	60	5	10	15	
7	14	21	11	13	17	19	7	28	22	23	26	27	11	7	14	29	13	31	33	34	7	17	21	11	19	37	31	7	14	13	22	33	23
38	41		43	44		46	47	49				51	52				53		54			56	57		58		59		61		62	63	

wherein blank segments contain any data.

143. (New) A method for transmitting data over a network to at least one client having a latency time to initiate transmission of said data to the client, including the steps of:

generating M anti-latency data streams containing at least a leading portion of data for receipt by the client, wherein said data is fragmented into K segments each requiring a time T to transmit over the network;

generating N interactive data streams containing at least a remaining portion of said data for the client to merge into after receiving at least a portion of an anti-latency data stream;

raising a request for said data;

connecting the client to the M anti-latency data streams and receiving data in the M anti-latency data streams;

pre-fetching at least a portion of data in the M anti-latency data streams in the client as pre-fetched data; and

refreshing the pre-fetched data during a refresh time period, wherein the refresh time period is 01:00-06:00,

and wherein: *K, M and N* are integers, and

T is a length of time.

144. (New) A method for transmitting data over a network to at least one client having a latency time to initiate transmission of said data to the client, including the steps of:

generating *M* anti-latency data streams containing at least a leading portion of data for receipt by the client, wherein said data is fragmented into *K* segments each requiring a time *T* to transmit over the network;

generating *N* interactive data streams containing at least a remaining portion of said data for the client to merge into after receiving at least a portion of an anti-latency data stream;

raising a request for said data;

connecting the client to the M anti-latency data streams and receiving data in the M anti-latency data streams;

pre-fetching at least a portion of data in the M anti-latency data streams in the client as pre-fetched data; and

refreshing the pre-fetched data during a refresh time period, wherein the refresh time period is 10:00-15:00,

and wherein: *K, M and N* are integers, and

T is a length of time.